Stochastic Online Scheduling with Precedence Constraints

Nicole Megow* Tjark Vredeveld †

June 8, 2007

Abstract

We consider the classical non-preemptive scheduling problem of processing jobs with precedence constraints on parallel machines with the objective to minimize the sum of (weighted) completion times. We discuss a reasonable online model and give lower and upper bounds on the competitive ratio for scheduling without job preemptions. These are the first investigation on online scheduling with precedence constraints for the considered objective function. We show matching bounds for scheduling on a single machine and corresponding bounds for scheduling on parallel machines.

Our results hold also in a the more general stochastic online scheduling model where, in addition to the limited information about the jobset, processing times are uncertain. In particular, we derive an n-approximation for the stochastic online scheduling problem $\mathrm{P}\,|\,\mathrm{prec}\,|\,\mathbb{E}\,[\,\sum w_j C_j\,]$. This bound is not constant but it is a first performance guarantee for non-preemptive stochastic scheduling that is independent of processing time distributions.

1 Introduction

One of the classical scheduling problems that has attracted research for decades is the problem of processing jobs with precedence constraints non-preemptively on parallel machines with the objective to minimize the sum of (weighted) completion times. We consider a stochastic online version of this problem where processing times are modelled as random variables and the jobs become known to the scheduler online.

In traditional online paradigms, i. e., the online-time and the online-list model [18, 24], it is assumed that all data about a request are revealed as soon as the request becomes known. Interpreted for an online scheduling problem with precedence constraints, this means that whenever a job arrives, a scheduler learns about its weight and processing time and – most importantly – about job dependencies. However, these dependencies occur between *two* jobs and it is not clear which job gets assigned the information about such a bilateral relation.

Certainly, there are various options to specify the information that should be revealed at job arrival. Nevertheless, we use a different model in which the moment of unveiling jobs and all their data is designated by other *job completions*: a scheduler learns about the existence of a job when all its predecessors have completed their processing. Then, its weight, processing time and all precedence relations to predecessors become known. One of the first publications we know of which applies this model, is by Feldmann, Kao, Sgall, and Teng [6].

Problem definition. Let $\mathcal{J} = \{1, 2, \ldots, n\}$ be a set of jobs which must be scheduled non-preemptively on m identical, parallel machines. Each of the machines can process at most one job at the time, and the jobs can be executed by any of the machines. All jobs must be scheduled in

*Technische Universität Berlin, Institut für Mathematik, Strasse des 17. Juni 136, 10623 Berlin, Germany. Email: nmegow@math.tu-berlin.de. Research supported by the DFG Research Center MATHEON "Mathematics for key technologies".

†Maastricht University, Department of Quantitative Economics, P.O. Box 616, 6200 MD Maastricht, The Netherlands. Email: t.vredeveld@ke.unimaas.nl. Research partially supported by METEOR, the Maastricht research school of Economics of Technology and Organizations.

compliance with the given precedence constraints. These constraints define a partial order $(\mathcal{J}, \prec)$ on the set of jobs $\mathcal{J}$, where $j \prec k$ implies that job k must not start processing before j has completed. If no precedence constraints are given, then we call the jobs independent.

Each job j must be processed for P_j units of time, where P_j is a positive random variable. By $\mathbb{E}[P_j]$ we denote the expected value of the processing time of job j and by p_j a particular realization of P_j. We assume that all random variables of processing times are stochastically independent. Additionally, each job j has associated a non-negative weight w_j. The goal is to find a non-anticipative scheduling policy [15] so as to minimize the total weighted completion time of the jobs, $\sum w_j C_j$, in expectation, where C_j denotes the completion time of job j. Adopting the well-known three-field classification scheme by Graham et al. [7] we denote the problem by $\mathrm{P}\,|\,prec\,|\,\mathbb{E}[\sum w_j C_j]$.

In this paper we consider the online version of this stochastic problem. As motivated above, we apply the following online paradigm: a job j becomes known to the scheduler when all its predecessors $k \prec j$ have completed their processing; at this point in time the weight w_j and the probability distribution of the processing time P_j are revealed. The solution of such a stochastic online scheduling problem is a non-anticipative, online scheduling policy; for more details we refer to [13, 11]. We aim for approximative policies, and as suggested in [13], we use a generalized definition of approximation guarantees from traditional stochastic offline scheduling [15]. Then, an (online) stochastic policy Π is a ρ-approximation, for some $\rho \geq 1$, if for all problem instances $\mathcal{I}$,

$$\mathbb{E}[\Pi(\mathcal{I})] \;\leq\; \rho\, \mathbb{E}[\textsc{Opt}(\mathcal{I})],$$

where $\mathbb{E}[\Pi(\mathcal{I})]$ and $\mathbb{E}[\textsc{Opt}(\mathcal{I})]$ denote the expected values that the policy Π and an optimal non-anticipatory offline policy, respectively, achieve on a given instance $\mathcal{I}$. The value ρ is called performance guarantee of policy Π.

Previous Work. The deterministic offline problem of scheduling jobs with precedence constraints to minimize the sum of (weighted) completion times has been shown to be NP-hard [9, 10] even if there is only a single processor. This classical problem, $1\,|\,prec\,|\sum w_j C_j$, has attracted research for more than thirty years and a vast amount of results has been obtained on this problem. Several classes of scheduling algorithms are known that achieve an approximation ratio of 2 in polynomial time whereas special cases are even solvable optimally; we refer to [4, 1] for a comprehensive overview. For scheduling on parallel machines, the currently best known approximation algorithm yields an approximation guarantee of $4 - 2/m$ and has been introduced by Munier et al. [17, 19]

Despite the obvious research interest in the scheduling problem under consideration, literature is very limited when assuming uncertainty in the problem data. The only related work we know of that deals with precedence constraints is by Skutella and Uetz [25] and deals with a stochastic offline scheduling model. The performance guarantees they prove are functions of a parameter Δ that bounds the squared coefficient of variation of processing times. Their policies require to solve a linear programming relaxation in which all jobs must be known in advance. This approach is not directly applicable in our online setting.

We are not aware of any other work on scheduling with precedence constraints to minimize the sum of completion times when information about the problem instance are incomplete. In particular, we do not know of any work on online models.

On the other hand, there has been done work on the deterministic online problem with respective to the makespan objective. Probably, one of the earliest publications using the online paradigm introduced above is by Feldmann et al. [6]. They consider scheduling of parallel jobs that are processed by more than one machine at the same time (malleable jobs). Considering jobs that are processed by at most one machine at the time, Azar and Epstein [2] derived a lower bound of $\Omega(\sqrt{m})$ for the competitive ratio of any deterministic or random online algorithm that schedules jobs (preemptively or not) on m related machines.

In our examination of the online environment it appeared that there exists relevant work on the deterministic offline problem of scheduling jobs with generalized precedence constraints, the so

	lower bound	upper bound	And/Or-prec [5]
$1 \mid prec \mid \mathbb{E}[\sum C_j]$	$\frac{2}{3}\sqrt{n}-1$	$\sqrt{2}\sqrt{n}$	$2\sqrt{n}$
$1 \mid prec \mid \mathbb{E}[\sum w_j C_j]$	$n-1$	n	n
$\mathrm{P} \mid prec \mid \mathbb{E}[\sum C_j]$	$\frac{2}{3}\sqrt{\frac{n}{m}}-1$	$\sqrt{2m}\sqrt{n}$	n
$\mathrm{P} \mid prec \mid \mathbb{E}[\sum w_j C_j]$	$\frac{n-1}{m}$	n	n

Table 1: Bounds on the performance guarantee of any stochastic online algorithm that does not use randomization [lower bound] and on the performance guarantee of S(E)PT [upper bound] for stochastic online scheduling of jobs with precedence constraints. Upper bounds for the unweighted setting are given for the special case where processing time distributions obey $\mathrm{Var}[P_j] \leq \mathbb{E}[P_j]^2$. Lower bounds for randomized algorithms can be derived, with value one half of the above bounds. Offline considerations in [5] for problems with And/Or-precedence relations transfer to our deterministic online setting and inspire our new results; their approximation guarantees are given in the third column.

called And/Or-precedence relations. While ordinary precedence constraints force a job to wait for the completion of *all* its predecessors (represented as an And-node in the corresponding precedence graph), there is an additional relaxed waiting condition that allows a job to start after at least one of its predecessors has completed (Or-node). Clearly, ordinary precedence constraints are contained as a special case in And/Or-precedence constraints. Erlebach, Kääb, and Möhring [5] derived approximation guarantees for the offline scheduling problem with generalized job dependencies which translate into competitiveness results for the deterministic online version of our problem. Their work also inspires our investigations in a stochastic online setting.

Our results. The contribution of this paper is twofold. We provide first results on online scheduling with precedence constraints to minimize the (weighted) sum of completion times. We derive matching upper and lower bounds for the problem on a single machine. We show that the basic Sept Algorithm achieves the best possible performance for this problem – albeit the competitive ratio in order of n (and $\sqrt{mn}$ when all job weights are equal) is discouraging. We also derive bounds for the corresponding problem on identical parallel machines which leave a gap in the order of the number of machines m. Our results are still valid when considering the more general And/Or-precedence constraints although this is not focus of our work. Thereby we improve the previously best known deterministic offline approximation result of n for the problem $\mathrm{P} \mid$ And/Or-prec $\mid \sum C_j$ by Erlebach et al. [5][1] to $\sqrt{2mn}$.

These results are not only the first ones for online scheduling with precedence constraints, they hold also in the more general model where jobs have stochastic processing times. Table 1 gives a summary of the lower and upper bounds we derived. In particular, we give an n-approximation for the problem $\mathrm{P} \mid \mathrm{prec} \mid \mathbb{E}[\sum w_j C_j]$. Even though this bound is non-constant it is of interest in the stochastic scheduling environment. In contrast to all previous non-preemptive stochastic (online) approximation results [16, 25, 13, 21] for scheduling with or without precedence constraints, our result is independent of the probability distributions of processing times.

2 Scheduling jobs with general weights

For scheduling independent jobs, good performance guarantees have been obtained for online versions of the classic Wspt algorithm, its stochastic counterpart Wespt, and various extensions [22, 12, 13, 11]. If jobs must obey precedence relations which are revealed after predecessor completion, no such variant yields a bounded performance. We give a simple single machine example where all jobs have even deterministic processing times. Note, that on a single machine no waiting time will reveal new information on the online sequence and is therefore superfluous.

[1]The approximation guarantee of n is actually shown for Spt on the single machine. The authors claim that the same bound holds for the parallel version of Spt.

Example 1. *Consider an instance that consists of the following three jobs. The first job has processing time $p_1 = k$ and weight $w_1 = 1$. Jobs 2 and 3 must obey the precedence constraint $2 \prec 3$; they have processing times $p_2 = 1$ and $p_3 = \varepsilon$, respectively, and their weights are $w_2 = \varepsilon$ and $w_3 = k$. Let k and ε be such that $\varepsilon << k$ and the ratios of weight over processing time of the two independent jobs 1 and 2 fulfill $w_1/p_1 > w_2/p_2$, that means $\varepsilon < 1/k$.*

Then the online version of the WSPT *algorithm schedules the jobs in increasing order of their indeces, $1, 2$, and 3 achieving an objective value of $k^2 + 2k + \varepsilon(2k + 1)$. In contrast, an optimal schedule has job 2 being processed first, followed by 3 and 1, and yields thus a value of $2k + 1 + \varepsilon(k + 2)$. If ε tends to 0, then the ratio of values of the* WSPT *schedule and an optimal schedule goes towards $\frac{k(k+2)}{2k+1}$ which is unbounded for increasing k.*

An even more basic online algorithm than WSPT is the online *Shortest Processing Time* (SPT) rule (or the online *Shortest Expected Processing Time* (SEPT) policy in a stochastic setting). Even though it seems counter intuitive to ignore known job weights, this algorithm yields a competitive ratio that matches the lower bound on the performance guarantee for any deterministic online algorithm on a single machine. In fact, Erlebach et al. [5] analyze the performance of the same algorithm in a deterministic offline setting with generalized job dependencies, the AND/OR-precedence constraints. The deterministic offline version of our problem with standard precedence constraints is a special case of the problem they consider. Moreover, their variant of the SPT algorithm considers only jobs that are *available* for processing according to the precedence constraints. Thus, it coincides with the online SPT algorithm that knows of jobs only after all predecessors have finished. Therefore, the approximation results translate into competitiveness results in our online setting if all processing times are deterministic.

Theorem 1 (Erlebach, Kääb, and Möhring [5])**.** *The online version of the* SPT *algorithm has a competitive ratio of n solving the scheduling problem $1\,|\,\mathrm{prec}\,|\sum w_j C_j$. If all jobs have equal weights, then* SPT *is $2\sqrt{n}$-competitive.*

This result is tight up to the constant factor of 2, see Theorems 5 and 7. Erlebach et al. [5] claim that the parallel machine version of SPT also yields an approximation guarantee of n for the weighted problem on parallel machines $\mathrm{P}\,|\,\mathrm{prec}\,|\sum w_j C_j$.

We extend these results to the more general setting in which all jobs have stochastic processing times without loosing in the performance guarantee. Consider the stochastic online scheduling problem $\mathrm{P}\,|\,\mathrm{prec}\,|\,\mathbb{E}\left[\sum w_j C_j\right]$ and the stochastic online policy that runs the SEPT policy on only *one out of m* machines. This policy simply ignores the $m - 1$ remaining available machines. We denote this algorithm by 1-SEPT.

Lemma 2. *The order of jobs in a schedule obtained by 1-*SEPT *is independent of the realization of processing times.*

Proof. We claim that for any two realizations of the processing times, at the completion of job j the same set of jobs is available, for any $j \in \mathcal{J}$. This implies the lemma, as 1-SEPT chooses the job to process only based on the set of available jobs and the expected processing times of these jobs.

To see the claim, consider two realizations of processing times. We assume that jobs are indexed in order in which they are processed in the first realization. First note that when no job has been processed, obviously the same set of jobs is available to 1-SEPT in both realizations. Suppose the claim is true up to job j. As 1-SEPT chooses job $j + 1$ to process after job j in the first realization and in the second realization the same set of jobs is available to 1-SEPT, the policy will also choose job $j + 1$ to be processed for the second realization. Hence, at the completion of job $j + 1$ the same set of jobs will be set free to 1-SEPT in the first as in the second realization. $\square$

We give a slightly different, stochastic version of the so-called *Threshold-Lemma* [5, Lemma 2] by Erlebach et al.

Let $\mathbb{E}\left[C_j^{\Pi}\right]$ denote the expected completion time of a job j in the schedule obtained by policy Π. Adopting the notation in [5], we define a stochastic version of the *threshold* ξ_j^{Π} of a job j

for policy Π as the maximum expected processing time of a job that finishes in expectation no later than j. More formally,

$$\xi_j^{\Pi} \;=\; \max_{k\in\mathcal{J}}\{\, \mathbb{E}\,[\, P_k \,] \mid \mathbb{E}\left[\, C_k^{\Pi} \,\right] \leq\; \mathbb{E}\left[\, C_j^{\Pi} \,\right]\}\,.$$

Thresholds have a useful property.

Lemma 3 (Threshold-Lemma)**.** *Let Π be a feasible policy for the stochastic scheduling problem $\mathrm{P}\,|\,prec\,|\,\mathbb{E}\,[\,\sum w_j C_j\,]$. Then for any job $j \in \mathcal{J}$ with threshold ξ_j^{Π} holds*

$$\xi_j^{1-\text{SEPT}} \;\leq\; \xi_j^{\Pi}\,.$$

Proof. For any non-anticipatory policy Π, we have that $\xi_j^{\Pi} \geq \mathbb{E}\,[\,P_j\,]$ since $\mathbb{E}\,[\,P_j\,] \leq \mathbb{E}\left[\,C_j^{\Pi}\,\right]$. If $\xi_j^{\Pi} = \mathbb{E}\,[\,P_j\,]$, then the Lemma holds. Suppose that $\xi_j^{\Pi} > \mathbb{E}\,[\,P_j\,]$. Then there exists a job k that was completed before job j by 1-Sept, that has expected processing time $\mathbb{E}\,[\,P_k\,] = \xi_j^{1-\text{SEPT}} > \mathbb{E}\,[\,P_j\,]$. As 1-Sept chooses the job with smallest expected processing time, we know by Lemma 2 that in any realization of processing times, job j cannot be available to 1-Sept when job k is started to be processed. Hence, k must be a predecessor of j. Thus, any policy processes job k before j, from which follows $\xi_j^{\Pi} \geq \mathbb{E}\,[\,P_k\,] = \xi_j^{1-\text{SEPT}}$. ◻

Now, we can establish a performance guarantee for the 1-Sept algorithm.

Theorem 4. *The* 1-Sept *algorithm that utilizes only one machine is an n-approximation for the stochastic online scheduling problem $\mathrm{P}\,|\,prec\,|\,\mathbb{E}\,[\,\sum w_j C_j\,]$.*

Proof. Let jobs be indexed in their order in the 1-Sept schedule. Recall from Lemma 2 that the order of jobs in the 1-Sept schedule is independent of the realization of processing times. Then $k < j$ implies $\mathbb{E}\left[\,C_k^{1-\text{SEPT}}\,\right] < \mathbb{E}\left[\,C_j^{1-\text{SEPT}}\,\right]$. Since there is no idle time, the expected completion time $\mathbb{E}\left[\,C_j^{1-\text{SEPT}}\,\right]$ of a job j in the 1-Sept schedule is

$$\mathbb{E}\left[\,C_j^{1-\text{SEPT}}\,\right] = \sum_{k=1}^{j} \mathbb{E}\,[\,P_k\,] \leq\; n\,\xi_j^{1-\text{SEPT}}\,. \tag{1}$$

With the Threshold-Lemma 3 and the fact that $\xi_j^{\Pi} \leq \mathbb{E}\left[\,C_j^{\Pi}\,\right]$ holds by definition for any policy Π – thus, also for an optimal policy Opt – we conclude from inequality (1)

$$\mathbb{E}\left[\,C_j^{\text{SEPT}}\,\right] \;\leq\; n\,\xi_j^{\text{OPT}} \;\leq\; n\,\mathbb{E}\left[\,C_j^{\text{OPT}}\,\right]\,.$$

Weighted summation over all jobs $j \in \mathcal{J}$ proves the theorem. ◻

In the following we show that no online algorithm using m machines can have a competitive ratio of less than $(n-1)/m$. Thus the analysis of the simple 1-Sept policy using one machine is tight if there is just one machine available, whereas in general it leaves a gap in the order of m. The lower bound is achieved even if the deterministic processing times are given. By definition, these bounds carry over to the more general stochastic online scheduling model.

Theorem 5. *No deterministic online algorithm can achieve a competitive ratio less than $(n-1)/m$ for the scheduling problem $\mathrm{P}\,|\,prec\,|\sum w_j C_j$ on any number of machines.*

Proof. Consider the following instance that consists of n jobs and assume, w.l.o.g., that $n-1$ is a multiple of the number of machines m. We have $n-1$ independent jobs $1, 2, \ldots, n-1$ with weights $w_j = 0$ and unit processing time. Suppose, that the online algorithm chooses the job ℓ to be scheduled as one of the last jobs (with maximum completion time). Then, we have one final job n in the instance with ℓ as its predecessor and with processing time zero and weight 1.

Clearly, an online algorithm can schedule the highly weighted last job only as the final job, achieving a schedule with value $(n-1)/m$. In contrast, an offline algorithm would choose job ℓ as one of the m first jobs to be processed, followed by the highly weighted job n. This yields value of 1. Thus, the ratio between both value is $(n-1)/m$. ◻

Adding a randomizing ingredient to the instance above, we extend the result to a lower bound for any randomized online algorithm. Here, we make use of Yao's principle [27].

Theorem 6. *The competitive ratio of any randomized online algorithm is bounded by $n/(2m)$ for the scheduling problem $\mathrm{P}\,|\,prec\,|\sum w_j C_j$ for any number of machines.*

Proof. Consider the instance in the previous proof. Against a randomized algorithm, the adversary does not know which will be the last scheduled job ℓ among the independent jobs. Therefore, we modify (and randomize) the instance by adding a random precedence relation between a job j, for $j = 1, \ldots, n-1$ and the job n with probability $1/(n-1)$, for any job j.

Clearly, an optimal offline solution yields a value of 1 as in the previous proof. Now, consider any deterministic online algorithm $\textsc{Alg}$. Let $\Pr[i \prec n]$ be the probability with that job i is the predecessor of job n. Any reasonable algorithm $\textsc{Alg}$ schedules the highly weighted job n immediately after it became known. Thus, the expected completion time of job n in the schedule of $\textsc{Alg}$ is

$$\mathbb{E}\left[C_n^{\textsc{Alg}}\right] \;\geq\; \sum_{i=1}^{n-1} \Pr[i \prec n] \cdot C_i \;=\; \frac{1}{n-1} \sum_{i=1}^{\frac{n-1}{m}} m \cdot i \;\geq\; \frac{n}{2m}\,.$$

Since all jobs $1, 2, \ldots, n-1$ have zero weight, their completion times do not contribute to the value of the schedule. Thus, the expected value of the schedule is

$$\mathbb{E}\left[\sum_j w_j C_j^{\textsc{Alg}}\right] \;=\; \mathbb{E}\left[C_n^{\textsc{Alg}}\right] \;\geq\; \frac{n}{2\,m}\,.$$

Hence, the ratio of the expected values of any deterministic schedule to the value of an optimal schedule is $n/(2m)$. By Yao's principle [27] this gives the lower bound for any online algorithm. $\square$

2.1 Scheduling Jobs with equal Weights

Notice that the lower bounds in the previous section heavily depend on choosing adequate job weights. Hence, they do not transfer to the problem setting where jobs have equal weights. For this relaxed problem $\mathrm{P}\,|\,prec\,|\sum C_j$, and thus also for its stochastic version, we show the following weaker lower bound which we complement by a strengthened upper bound matching up to a constant factor for certain processing time distributions.

Theorem 7. *The competitive ratio of any deterministic online algorithm for the scheduling problem $\mathrm{P}\,|\,prec\,|\sum C_j$ has a lower bound of $\frac{2}{3}\sqrt{n/m} - 1$. A lower bound for randomized online algorithms is $\frac{1}{3}\sqrt{n/m}$.*

Proof. First, we show the lower bound for deterministic online algorithms. We have mk independent jobs with processing times $p_j = 1$ for all $j = 1, \ldots, mk$ where $k >> m$. Moreover, there are $mk^2 - mk$ jobs that have length 0 and which must obey precedence constraints that form one long chain $mk+1 \prec mk+2 \prec \ldots$. Let $\ell \in \{1, \ldots, mk\}$ be the job to be scheduled last by the online algorithm among the independent jobs. This job ℓ is a predecessor of $mk+1$, the first job of the chain. Note, that an online algorithm $\textsc{Alg}$ cannot start the job chain with $mk^2 - mk$ jobs earlier than time k. $\textsc{Alg}$ yields a schedule of value

$$\textsc{Alg} \;\geq\; m \sum_{i=1}^{k} i + (mk^2 - mk)k \;=\; \frac{1}{2}\, mk\,(1 + (2k-1)\,k) \;\geq\; \frac{1}{2}\, mk^2 (2k-1)\,. \tag{2}$$

In contrast, an optimal offline algorithm $\textsc{Opt}$ knows the sequence in advance. By processing job ℓ at time 0 and starting the chain of zero length jobs at time 1, it can achieve an objective value

$$\textsc{Opt} \;\leq\; m \sum_{i=1}^{k} i + mk^2 - mk \;=\; \frac{1}{2}\, mk\,(3k-1) \;\leq\; \frac{3}{2}\, mk^2\,. \tag{3}$$

The ratio of the bounds in (2) and (3) combined with the number of jobs, $n = mk^2$, gives the lower bound on the competitive ratio of any deterministic online algorithm.

$$\frac{\textsc{Alg}}{\textsc{Opt}} \geq \frac{2k-1}{3} > \frac{2}{3}\sqrt{\frac{n}{m}} - 1 .$$

The proof that one half of this value is a lower bound for randomized online algorithms is similar to the proof of Theorem 6. We consider the instance above and replace the precedence constraint $\ell \prec mk+1$ by a randomized variant. That means, with probability $\Pr[j \prec k+1] = 1/(mk)$, a job $j = 1, \ldots, mk$ precedes job $mk+1$, the first job of the job chain. $\square$

Clearly, the upper bound of n on the performance guarantee for the 1-Sept algorithm in Theorem 4 holds in the unweighted setting considered here. Below we derive a performance bound which is improved for a large class of problem instances. In particular, for instances with bounded processing time variation our result improves on the previous n approximation (Theorem 4). We achieve this bound by extending ideas of Erlebach et al. [5] for the deterministic single-machine setting and combining them with a lower bound on the expected optimal value for the relaxed problem $\mathrm{P}\,|\,|\,\mathbb{E}\left[\sum C_j\right]$ given by Möhring, Schulz, and Uetz [16].

Lemma 8 (Möhring et al. [16])**.** *Given an instance of* $\mathrm{P}\,|\,|\,\mathbb{E}\left[\sum C_j\right]$ *with jobs indexed in non-decreasing order of expected processing times* $\mathbb{E}\left[P_j\right]$*, an optimal policy* Opt *yields a value*

$$\sum_j \mathbb{E}\left[C_j^{\textsc{Opt}}\right] \geq \sum_j \sum_{k=1}^{j} \frac{\mathbb{E}\left[P_k\right]}{m} - \frac{(m-1)(\Delta-1)}{2m} \sum_j \mathbb{E}\left[P_j\right] ,$$

where Δ *bounds the squared coefficient of variation of the processing times, that is,* $\mathrm{Var}[P_j]/\mathbb{E}\left[P_j\right]^2 \leq \Delta$ *for all jobs* $j = 1, \ldots, n$ *and some* $\Delta \geq 0$*.*

Theorem 9. *The 1-*Sept *policy achieves an approximation guarantee of*

$$\rho = \frac{1}{2}(m-1)(\Delta-1) + \frac{1}{2}\sqrt{[(m-1)(\Delta-1)]^2 + 8mn} ,$$

with $\Delta \geq \mathrm{Var}[P_j]/\mathbb{E}\left[P_j\right]^2$ *for any instance of the stochastic online problem* $\mathrm{P}\,|\,\mathrm{prec}\,|\,\mathbb{E}\left[\sum C_j\right]$*.*

Proof. Consider an 1-Sept schedule and re-index jobs such that their index corresponds to its position in the 1-Sept-order. Let $\alpha > (m-1)(\Delta-1)/\sqrt{n}$ be a parameter that will be specified later. Let x be the last job (with maximum index) in the 1-Sept schedule such that all jobs with a smaller index have an expected processing time of at most $\mathbb{E}\left[C_x^{\textsc{Sept}}\right]/(\alpha\sqrt{n})$; that is, more formally

$$x := \max\left\{ j \in \mathcal{J} \,|\, \mathbb{E}\left[P_k\right] \leq \frac{\mathbb{E}\left[C_j^{\textsc{Sept}}\right]}{\alpha\sqrt{n}} \text{ for all } k \leq j \right\} .$$

This designated job, x, is used to partition the set of jobs into two disjunctive subsets: $\mathcal{J}^{\leq}$ denotes the set of jobs that complete before x in the 1-Sept schedule, i. e., $\mathcal{J}^{\leq} = \{j \in \mathcal{J} \,|\, j \leq x\}$, and $\mathcal{J}^{>}$ consists of the remaining jobs $\mathcal{J} \setminus \mathcal{J}^{\leq}$. Obviously, the expected completion time of job x is $\mathbb{E}\left[C_x^{1-\textsc{Sept}}\right] = \sum_{j \in \mathcal{J}^{\leq}} \mathbb{E}\left[P_j\right]$. Now, the expected value of the Sept schedule can be expressed as

$$\sum_{j \in \mathcal{J}} \mathbb{E}\left[C_j^{1-\textsc{Sept}}\right] = \sum_{j \in \mathcal{J}^{\leq}} \mathbb{E}\left[C_j^{1-\textsc{Sept}}\right] + \sum_{j \in \mathcal{J}^{>}} \mathbb{E}\left[C_j^{1-\textsc{Sept}}\right] .$$

We bound the expected completion times of jobs of both job sets separately. To bound the contribution of the jobs in $\mathcal{J}^{\leq}$, assume that $\mathcal{J}^{\leq} \neq \emptyset$.

Let Opt be an optimal policy for all jobs $j \in \mathcal{J}$ and Opt' an optimal policy that schedules only the jobs in $\mathcal{J}^{\leq}$. Clearly,

$$\sum_{j \in \mathcal{J}^{\leq}} \mathbb{E}\left[C_j^{\textsc{Opt}}\right] \geq \sum_{j \in \mathcal{J}^{\leq}} \mathbb{E}\left[C_j^{\textsc{Opt}'}\right] . \tag{4}$$

By ignoring the release dates, we can use Lemma 8. Assuming that the jobs in $\mathcal{J}^{\le}$ are indexed, $1,\dots,|\mathcal{J}^{\le}|$, in non-decreasing order of their expected processing times, we obtain:

$$\sum_{j\in\mathcal{J}^{\le}} \mathbb{E}\left[C_j^{\textsc{Opt}'}\right] \quad\ge\quad \sum_{j\in\mathcal{J}^{\le}}\sum_{k=1}^{j}\frac{\mathbb{E}\left[P_k\right]}{m} \;-\; \frac{(m-1)(\Delta-1)}{2m}\sum_{j\in\mathcal{J}^{\le}}\mathbb{E}\left[P_j\right], \tag{5}$$

where $\Delta \ge \mathrm{Var}[P_j]/\mathbb{E}\left[P_j\right]^2$ for all jobs $j\in\mathcal{J}$ and some $\Delta\ge 0$. We claim that $\sum_{j\in\mathcal{J}^{\le}}\sum_{k=1}^{j}\mathbb{E}\left[P_k\right]$ is bounded from below by

$$\frac{\alpha\sqrt{n}}{2}\,\mathbb{E}\left[C_x^{1-\textsc{Sept}}\right].$$

To see this claim, note that

$$\sum_{j\in\mathcal{J}^{\le}}\sum_{k=1}^{j}\mathbb{E}\left[P_k\right] = \sum_{k\in\mathcal{J}^{\le}}\left(|\mathcal{J}^{\le}|-k+1\right)\mathbb{E}\left[P_k\right].$$

This value can not be less than the minimum of this value over all possible expected processing times for jobs in $j\in\mathcal{J}^{\le}$ satisfying $\sum_{j\in\mathcal{J}^{\le}}\mathbb{E}\left[P_j\right] = \mathbb{E}\left[C_x^{1-\textsc{Sept}}\right]$ and $\mathbb{E}\left[P_j\right] \le \mathbb{E}\left[C_x^{1-\textsc{Sept}}\right]/(\alpha\sqrt{n})$. This minimum is obviously obtained by setting $\mathbb{E}\left[P_j\right] = \mathbb{E}\left[C_x^{1-\textsc{Sept}}\right]/(\alpha\sqrt{n})$ for $j=|\mathcal{J}^{\le}|-b+1,\dots,|\mathcal{J}^{\le}|$, $\mathbb{E}\left[P_j\right]=(1-b/(\alpha\sqrt{n}))\mathbb{E}\left[C_x^{1-\textsc{Sept}}\right]$, for $j=|\mathcal{J}^{\le}|-b$, and $\mathbb{E}\left[P_j\right]=0$ for all other j, where $b=\lfloor\alpha\sqrt{n}\rfloor$. This proves the claim and with $\sum_{j\in\mathcal{J}^{\le}}\mathbb{E}\left[P_j\right]=\mathbb{E}\left[C_x^{1-\textsc{Sept}}\right]$ we have

$$\begin{aligned}\sum_{j\in\mathcal{J}^{\le}}\mathbb{E}\left[C_j^{\textsc{Opt}}\right] \quad&\ge\quad \frac{\alpha\sqrt{n}}{2m}\mathbb{E}\left[C_x^{1-\textsc{Sept}}\right] - \frac{(m-1)(\Delta-1)}{2m}\mathbb{E}\left[C_x^{1-\textsc{Sept}}\right]\\ &=\quad \frac{\alpha\sqrt{n}-(m-1)(\Delta-1)}{2m}\mathbb{E}\left[C_x^{1-\textsc{Sept}}\right].\end{aligned}$$

With this estimate of the relevant portion of the expected optimal value we can bound the value achieved by the 1-Sept policy

$$\sum_{j\in\mathcal{J}^{\le}}\mathbb{E}\left[C_j^{1-\textsc{Sept}}\right] \;\le\; n\,\mathbb{E}\left[C_x^{1-\textsc{Sept}}\right] \;\le\; \frac{2\,m\,n}{\alpha\sqrt{n}-(m-1)(\Delta-1)}\sum_{j\in\mathcal{J}^{\le}}\mathbb{E}\left[C_j^{\textsc{Opt}}\right], \tag{6}$$

if and only if $\alpha\sqrt{n} > (m-1)(\Delta-1)$.

Consider now jobs in the remaining job set $\mathcal{J}^{>}$; by definition, there exists for each job $j\in\mathcal{J}^{>}$ a job k that completes in 1-Sept earlier than j and has processing time $\mathbb{E}\left[P_k\right] > \mathbb{E}\left[C_j^{1-\textsc{Sept}}\right]/(\alpha\sqrt{n})$. We conclude from this fact and the Threshold-Lemma 3 (including the notion of the threshold ξ_j^S) that for all $j\in\mathcal{J}^{>}$ holds

$$\mathbb{E}\left[C_j^{\textsc{Opt}}\right] \;\ge\; \xi_j^{\textsc{Opt}} \;\ge\; \xi_j^{1-\textsc{Sept}} \;\ge\; \mathbb{E}\left[P_k\right] \;>\; \frac{\mathbb{E}\left[C_j^{1-\textsc{Sept}}\right]}{\alpha\sqrt{n}}.$$

Summation over all jobs $j\in\mathcal{J}^{>}$ yields a bound on the completion times in the 1-Sept schedule,

$$\sum_{j\in\mathcal{J}^{>}}\mathbb{E}\left[C_j^{1-\textsc{Sept}}\right] \;\le\; \alpha\sqrt{n}\sum_{j\in\mathcal{J}^{>}}\mathbb{E}\left[C_j^{\textsc{Opt}}\right].$$

Finally, combination with Equality (6) yields the bound

$$\sum_{j\in\mathcal{J}}\mathbb{E}\left[C_j^{1-\textsc{Sept}}\right] \;\le\; \max\left\{\frac{2\,m\,n}{\alpha\sqrt{n}-(m-1)(\Delta-1)},\;\alpha\sqrt{n}\sum_{j\in\mathcal{J}}\mathbb{E}\left[C_j^{\textsc{Opt}}\right]\right\}.$$

The performance bound is minimized when choosing the parameter $\alpha := ((m-1)(\Delta-1) + \sqrt{[(m-1)(\Delta-1)]^2 + 8mn})/(2\sqrt{n})$ which gives the desired approximation guarantee

$$\rho = \frac{1}{2}(m-1)(\Delta-1) + \frac{1}{2}\sqrt{[(m-1)(\Delta-1)]^2 + 8mn}\,.$$

Observe that the optimal choice of α fulfills the condition $\alpha\sqrt{n} > (m-1)(\Delta-1)$ in equality (6). $\square$

In contrast to the previous, more general approximation guarantee of value n (Theorem 4), this result depends again on the variance of processing times. In particular, ρ grows with the parameter $\Delta \geq \mathrm{Var}[P_j]/\mathbb{E}[P_j]^2$ for all jobs. However, for instances with distributions of small relative variance, this bound improves on the n-approximation for the general weighted problem in Theorem 4. More precisely, for

$$\Delta \leq \frac{n}{m-1} - 1$$

the performance guarantee ρ is at most n.

Theorem 9 leads immediately to the following result for a restricted class of probability distributions – the NBUE distributions (new better than used in expectation)[2], which imply $\Delta \leq 1$ [8].

Corollary 10. *The 1-*Sept *algorithm that utilizes only one machine is $\sqrt{2mn}$-competitive for the stochastic online scheduling problem* $\mathrm{P}\,|\,prec\,|\,\mathbb{E}\left[\sum C_j\right]$ *if all jobs have processing times that follow a NBUE distribution, i. e., $\Delta \leq 1$.*

This performance guarantee improves the general bound $\rho = n$ for scheduling with individual job weights in Theorem 4 iff the number of jobs n is larger than $4m^2$. Furthermore, it follows from the analysis that the result hold also if And/Or-precedence constraints are present. Thus, we improve the approximation factor of n given for the offline scheduling problem where jobs have equal weights and processing times are deterministic in [5] even though we consider a more general model.

3 Conclusion & Open Question

We presented first results for (stochastic) online scheduling with precedence constraints to minimize the (expected) sum of weighted completion times. The bounds for the single machine setting are tight whereas in the parallel machine setting is left a gap of factor m.

For the problem $\mathrm{P}\,|\,prec\,|\,\mathbb{E}\left[\sum w_j C_j\right]$ we give an n-approximation. Even though this result is discouraging at the first look since it is non-constant, it is interesting when comparing it to other non-preemptive stochastic approximation results [16, 25, 13, 21]. In contrast to all previous results, our bound does not depend on the distribution of processing times. All previous results have a performance guarantee that depends on an upper bound Δ on the squared coefficient of variation of processing times. The reason is that all results are obtained using the same lower bound on the expected value of an optimal policy that was derived by Möhring, Schulz, and Uetz in [16]. It is based on a stochastic version of a linear programming relaxation by Schulz [20] which contains information about the processing time distributions other than the expected values. In particular, the upper bound Δ plays a crucial role.

An open question is if there exist policies with a constant performance guarantee independent of processing time distributions. The n-approximation in this paper could be interpreted as a motivating step towards a positive answer of this question.

The fundamental problem, as we see it, is that until recently the bound by Möhring et al. [16] was the only known non-trivial lower bound on the value of an optimal non-preemptive policy. However, recently there has been introduced in [14] a new lower bound on the optimal expected value for the preemptive version of the scheduling problem. This bound is derived by borrowing ideas for a fast single machine relaxation from Chekuri et al. [3]. The crucial ingredient to the

[2]Examples of NBUE distributions are exponential, Erlang, uniform, or Weibull distributions (the latter with shape parameter at least 1).

result is then the application of a Gittins index priority policy [23, 26] which is optimal to a relaxed version of the fast single machine relaxation. For preemptive scheduling on parallel machines, a constant performance guarantee of 2 independent of the probability distributions is shown even in an online setting [14].

Clearly, this new bound also holds as a lower bound for the expected value of a non-preemptive optimal policy. Even though it is not clear how to apply it within the analysis of non-preemptive policies, we believe that the Gittins-index based lower bound may lead to new, possibly constant performance guarantees for non-preemptive scheduling.

References

[1] C. Ambühl and M. Mastrolilli. Single machine precedence constrained scheduling is a vertex cover problem. In Y. Azar and T. Erlebach, editors, *Proceedings of 14th European Symposium on Algorithms*, number 4168 in Lecture Notes in Computer Science, pages 28–39, Zurich, Switzerland, 2006. Springer.

[2] Y. Azar and L. Epstein. On-line scheduling with precedence constraints. *Discrete Applied Mathematics*, 119:169–180, 2002.

[3] C. Chekuri, R. Motwani, B. Natarajan, and C. Stein. Approximation techniques for average completion time scheduling. *SIAM Journal on Computing*, 31:146–166, 2001.

[4] J. R. Correa and A. S. Schulz. Single-machine scheduling with precedence constraints. *Mathematics of Operations Research*, 30(4):1005–1021, 2005.

[5] T. Erlebach, V. Kääb, and R. H. Möhring. Scheduling AND/OR-networks on identical parallel machines. In K. Jansen and R. Solis-Oba, editors, *Proceedings of the First International Workshop on Approximation and Online Algorithms, WAOA 2003*, volume 2909 of *Lecture Notes in Computer Science*, pages 123–136, Budapest, Hungary, 2004. Springer.

[6] A. Feldmann, M.-Y. Kao, J. Sgall, and S.-H. Teng. Optimal online scheduling of parallel jobs with dependencies. *Journal of Combinatorial Optimization*, 1(4):393–411, 1998.

[7] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, 5:287–326, 1979.

[8] W. J. Hall and J. A. Wellner. Mean residual life. In M. Csörgö, D. A. Dawson, J. N. K. Rao, and A. K. Md. E. Saleh, editors, *Proceedings of the International Symposium on Statistics and Related Topics*, pages 169–184, Ottawa, ON, Canada, 1981.

[9] E. L. Lawler. Sequencing jobs to minimize total weighted completion time subject to precedence constraints. *Annals of Discrete Mathematics*, 2:75–90, 1978.

[10] J. K. Lenstra and A. H. G. Rinnooy Kan. Complexity of scheduling under precedence constraints. *Operations Research*, 26(1):22–35, 1978.

[11] N. Megow. *Coping with incomplete information in scheduling – stochastic and online models.* Dissertation, Technische Universität Berlin, 2006.

[12] N. Megow and A. S. Schulz. On-line scheduling to minimize average completion time revisited. *Operations Research Letters*, 32(5):485–490, 2004.

[13] N. Megow, M. Uetz, and T. Vredeveld. Models and algorithms for stochastic online scheduling. *Mathematics of Operations Research*, 31(3):513–525, 2006.

[14] N. Megow and T. Vredeveld. Approximation in preemptive stochastic online scheduling. In Y. Azar and T. Erlebach, editors, *Proceedings of 14th European Symposium on Algorithms*, number 4168 in Lecture Notes in Computer Science, pages 516–527, Zurich, Switzerland, 2006. Springer.

[15] R. H. Möhring, F. J. Radermacher, and G. Weiss. Stochastic scheduling problems I: General strategies. *Zeitschrift für Operations Research*, 28:193–260, 1984.

[16] R. H. Möhring, A. S. Schulz, and M. Uetz. Approximation in stochastic scheduling: the power of LP-based priority policies. *Journal of the ACM*, 46:924–942, 1999.

[17] A. Munier, M. Queyranne, and A. S. Schulz. Approximation bounds for a general class of precedence constrained parallel machine scheduling problems. In R. Bixby, E. Boyd, and R. R. Mercado, editors, *Proceedings of the 6th Mathematical Programming Society Conference on Integer Programming and Combinatorial Optimization*, volume 1412 of *Lecture Notes in Computer Science*, pages 367–382, Houston, Texas, USA, 1998. Springer.

[18] K. R. Pruhs, J. Sgall, and E. Torng. Online scheduling. In J. Y.-T. Leung, editor, *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, chapter 15. Chapman & Hall/CRC, 2004.

[19] M. Queyranne and A. S. Schulz. Approximation bounds for a general class of precedence constrained parallel machine scheduling problems. *SIAM Journal on Computing*, 35(5):1241–1253, 2006.

[20] A. S. Schulz. Scheduling to minimize total weighted completion time: Performance guarantees of LP-based heuristics and lower bounds. In W. H. Cunningham, S. T. McCormick, and M. Queyranne, editors, *Proceedings of the 5th Mathematical Programming Society Conference on Integer Programming and Combinatorial Optimization*, volume 1084 of *Lecture Notes in Computer Science*, pages 301–315, Vancouver, BC, Canada, 1996. Springer.

[21] A. S. Schulz. New old algorithms for stochastic scheduling. In S. Albers, R. H. Möhring, G. C. Pflug, and R. Schultz, editors, *Algorithms for Optimization with Incomplete Information*, number 05031 in Dagstuhl Seminar Proceedings, 2005.

[22] A. S. Schulz and M. Skutella. The power of α-points in preemptive single machine scheduling. *Journal of Scheduling*, 5:121–133, 2002.

[23] K. C. Sevcik. Scheduling for minimum total loss using service time distributions. *Journal of the ACM*, 21:65–75, 1974.

[24] J. Sgall. On-line scheduling – a survey. In A. Fiat and G. J. Woeginger, editors, *Online Algorithms: The State of the Art*, volume 1442 of *Lecture Notes in Computer Science*, pages 196–231. Springer, Berlin, 1998.

[25] M. Skutella and M. Uetz. Stochastic machine scheduling with precedence constraints. *SIAM Journal on Computing*, 34(4):788–802, 2005.

[26] G. Weiss. On almost optimal priority rules for preemptive scheduling of stochastic jobs on parallel machines. *Advances in Applied Probability*, 27:827–845, 1995.

[27] A. C.-C. Yao. Probabilistic computations: toward a unified measure of complexity (extended abstract). In *Proceedings of the 18th IEEE Symposium on the Foundations of Computer Science*, pages 222–227, 1977.